BDA_Lab1

Olayemi Morrison (olamo208) and Greeshma Jeev Koothuparambil(greko370)

2024-05-16

Question 1

What are the lowest and highest temperatures measured each year for the period 1950-2014. Provide the lists sorted in the descending order with respect to the maximum temperature. In this exercise you will use the temperature-readings.csv file.

```python
from pyspark import SparkContext
sc = SparkContext(appName = "Lab")
temp_reading = sc.textFile("BDA/input/temperature-readings.csv")
lines = temp_reading.map(lambda line: line.split(";"))

# (key, value) = (year,temperature)
year_temp = lines.map(lambda x: (x[1][0:4], float(x[3])))

#filter
year_temp = year_temp.filter(lambda x: int(x[0])>=1950 and int(x[0])<=2014)

#Get max
max_temp = year_temp.reduceByKey(lambda a,b: max(a, b))
max_temp = max_temp.sortBy(ascending = False, keyfunc=lambda k: k[1])

#Get min
min_temp = year_temp.reduceByKey(lambda a,b: min(a, b))
min_temp = min_temp.sortBy(ascending = False, keyfunc=lambda k: k[1])

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
max_temp.saveAsTextFile("BDA/output/max")
min_temp.saveAsTextFile("BDA/output/min")
```

Result

Year, temperature
min:
('1990', -35.0)
('1952', -35.5)
('1974', -35.6)
('1954', -36.0)
…

max:
('1975', 36.1)
('1992', 35.4)
('1994', 34.7)

Question 2.1

Count the number of readings for each month in the period of 1950-2014 which are higher than 10 degrees.

```
temp_reading = sc.textFile("BDA/input/temperature-readings.csv")
lines = temp_reading.map(lambda line: line.split(";"))
# (key, value) = ((year,month),temperature)
year_temp = lines.map(lambda x: ((x[1][0:4],x[1][5:7]), float(x[3])))

#filter
year_temp = year_temp.filter(lambda x: int(x[0][0])>=1950 and int(x[0][0])<=2014 and x[1]>10)

#(key, value) = ((year,month),count)
year_month_count=year_temp.map(lambda x:(x[0],1))

#Get counts
year_month_counts = year_month_count.reduceByKey(lambda a,b: a+1)

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
year_month_counts.saveAsTextFile("BDA/output")
```

Result

Year, month, count
((‘2008’, ‘11’), 203)
((‘1971’, ‘06’), 2033)
((‘1989’, ‘12’), 4)
((‘1966’, ‘11’), 46)
((‘1956’, ‘05’), 574)
((‘1998’, ‘07’), 6048)
((‘1975’, ‘02’), 10)
((‘1983’, ‘10’), 289)

Question 2.2

Repeat the exercise, this time taking only distinct readings from each station. That is, if a station reported a reading above 10 degrees in some month, then it appears only once in the count for that month.

```
# (key, value) = ((year,month,station),temperature)
year_temp = lines.map(lambda x: ((x[1][0:4],x[1][5:7],x[0]), float(x[3])))\
                    .filter(lambda x: int(x[0][0])>=1950 and int(x[0][0])<=2014 and x[1]>10)

# Reduce the value
year_month_sta_count=year_temp.reduceByKey(lambda x,y:max(x,y))\
                    .map(lambda x:((x[0][0],x[0][1]),x[1]))\
                    .countByKey()         #Count the number of each key and return a dict

sc.parallelize(list(year_month_sta_count.items())).saveAsTextFile("BDA/output")
```

Results

Year, month, count
((‘2010’, ‘05’), 319)
((‘1957’, ‘01’), 2)
((‘1998’, ‘10’), 228)
((‘1959’, ‘09’), 127)

```
(('1999', '11'), 225)
(('1988', '04'), 220)
(('1960', '07'), 126)
(('1989', '05'), 318)
(('2004', '09'), 321)
(('1994', '03'), 89)
(('2008', '04'), 296)
(('1985', '07'), 304)
(('2001', '04'), 230)
(('1995', '06'), 294)
(('1951', '08'), 112)
(('1973', '06'), 377)
(('1950', '09'), 50)
(('1971', '12'), 27)
```

Question 3

Find the average monthly temperature for each available station in Sweden. Your result should include average temperature for each station for each month in the period of 1960-2014.

```python
# (key, value) = ((year,month,date,station),temperature)
year_temp = lines.map(lambda x: ((x[1][0:4],x[1][5:7],x[1][8:],x[0]), float(x[3])))\
                    .filter(lambda x: int(x[0][0])>=1960 and int(x[0][0])<=2014)

# Compute the month average temperature for each station
#daily max temperature
daily_max_temp=year_temp.reduceByKey(lambda x,y:max(x,y))
#daily min temperature
daily_min_temp=year_temp.reduceByKey(lambda x,y:min(x,y))
#Average monthly temperature
avg_monthly_temp=daily_max_temp.join(daily_min_temp)\
                        .map(lambda x:((x[0][0],x[0][1],x[0][3]),(x[1][0]+x[1][1])/2))\
                        .groupByKey()\
                        .mapValues(lambda x:sum(x)/len(x))

print(avg_monthly_temp.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
avg_monthly_temp.saveAsTextFile("BDA/output")
```

Result

Year, month, station number, average monthly temperature

```
(('1989', '06', '92400'), 14.686666666666667)
(('1982', '09', '107530'), 11.171666666666667)
(('2002', '11', '136360'), -5.861666666666668)
(('1964', '04', '53370'), 8.046666666666667)
(('1967', '08', '98170'), 15.408064516129032)
(('2002', '08', '181900'), 15.598387096774191)
(('1996', '08', '96190'), 17.099999999999998)
(('1980', '05', '155940'), 3.8870967741935485)
```

Question 4

Provide a list of stations with their associated maximum measured temperatures and maximum measured daily precipitation. Show only those stations where the maximum temperature is between 25 and 30 degrees and maximum daily precipitation is between 100 mm and 200 mm.

```
temp_reading = sc.textFile("BDA/input/temperature-readings.csv")
precipitation_file=sc.textFile("BDA/input/precipitation-readings.csv")
temperature_lines = temp_reading.map(lambda line: line.split(";"))
precipitation_lines = precipitation_file.map(lambda line: line.split(";"))

#Temperature
# (key, value) = (station,temperature)
station_temperature = temperature_lines.map(lambda x: (x[0], float(x[3])))\
                      .reduceByKey(lambda x,y:max(x,y))\
                      .filter(lambda x:x[1]>25 and x[1]<30)

#Precipitation
# (key, value) = ((year,month,date,station),precipitation)
station_precipitation= precipitation_lines.map(lambda x: ((x[1][0:4],x[1][5:7],
x[1][8:],x[0]), float(x[3])))\
                      .reduceByKey(lambda x,y:x+y)\
                      .map(lambda x:(x[0][3],x[1]))\
                      .reduceByKey(lambda x,y:max(x,y))\
                      .filter(lambda x:x[1]>100 and x[1]<200)

#Join
conbine_data=station_temperature.join(station_precipitation)

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
conbine_data.saveAsTextFile("BDA/output")
```

Result

Station number, maximum measured temperature, maximum daily precipitation

None found.

Question 5

Calculate the average monthly precipitation for the Östergotland region (list of stations is provided in the separate file) for the period 1993-2016. In order to do this, you will first need to calculate the total monthly precipitation for each station before calculating the monthly average (by averaging over stations).

```
precipitation_file=sc.textFile("BDA/input/precipitation-readings.csv")
ost_station_file=sc.textFile("BDA/input/stations-Ostergotland.csv")
precipitation_lines = precipitation_file.map(lambda line: line.split(";"))
ost_station_lines_bc=ost_station_file.map(lambda line: line.split(";")).map(lambda x:int(x[0]))

# broadcast
bc=sc.broadcast(ost_station_lines_bc.collect())

# (key, value) = ((year,month,station),precipitation)
ost_average_monthly_precipitation=precipitation_lines.filter(lambda x:(int(x[0])in bc.value) and \
                                    int(x[1][0:4])>=1993 and int(x[1][0:4])<=2016)\
        .map(lambda x:((x[1][0:4],x[1][5:7],x[0]),float(x[3])))\
        .reduceByKey(lambda x,y:x+y)\
        .map(lambda x:((x[0][0],x[0][1]),x[1]))\
        .groupByKey()\
        .mapValues(lambda x:sum(x)/len(x))

print(ost_average_monthly_precipitation.collect())

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
ost_average_monthly_precipitation.saveAsTextFile("BDA/output")
```

Result

Year, month, average monthly precipitation
((‘2012’, ‘09’), 72.75)
((‘1995’, ‘05’), 26.00000000000002)
((‘2011’, ‘08’), 86.26666666666665)
((‘2001’, ‘02’), 36.7666666666667)
((‘2007’, ‘06’), 108.94999999999999)
((‘2011’, ‘06’), 88.35000000000001)
((‘2011’, ‘10’), 43.75000000000001)
((‘2014’, ‘10’), 72.13749999999999)
((‘1996’, ‘09’), 57.46666666666667)
…